NASA CR-134471

NASIS DATA BASE MANAGEMENT SYSTEM - IBM 360/370 OS MVT IMPLEMENTATION

II - OVERVIEWS

NEOTERICS, INC.

prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

NASA Lewis Research Center
Contract NAS 3-14979

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-134471		

4. Title and Subtitle	5. Report Date
NASIS DATA BASE MANAGEMENT SYSTEM - IBM 360/370 OS MVT IMPLEMENTATION II - OVERVIEWS	September 1973
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No. None
	10. Work Unit No.
9. Performing Organization Name and Address Neoterics, Inc. 2800 Euclid Avenue Cleveland, Ohio 44115	11. Contract or Grant No. NAS 3-14979
	13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

Final Report. Project Manager, Charles M. Goldstein, Computer Services Division, NASA Lewis Research Center, Cleveland, Ohio

16. Abstract

The NASIS development workbook contains all the required system documentation. The workbook includes the following seven volumes:

I - Installation Standards (CR-134470)
II - Overviews (CR-134471)
III - Data Set Specifications (CR-134472)
IV - Program Design Specifications (CR-134473)
V - Retrieval Command System Reference Manual (CR-134474)
VI - NASIS Message File (CR-134475)
VII - Data Base Administrator User's Guide (CR-134476)

17. Key Words (Suggested by Author(s))	18. Distribution Statement
	Unclassified - unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages
Unclassified	Unclassified	37

* For sale by the National Technical Information Service, Springfield, Virginia 22151

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

INTRODUCTION

The NASA Aerospace Safety Information System (NASIS) was developed by Neoterics, Inc. of Cleveland, Ohio, to satisfy the needs of the Aerospace Safety Research and Data Institute (ASRDI) for the management of safety related information. However, since its conception, NASIS has evolved into an interactive generalized data base management system. While sounding impressive, what is really meant by these terms prefixing "system"?

It is the intent of this document to define these terms as they relate to NASIS by providing an overview of the latest version of the system and its capabilities.

I. THE SYSTEM

NASIS is an interactive generalized data base management system. This section will define these identifying terms in relation to NASIS, and will introduce the functional components of the system.

A. INTERACTIVE SYSTEM

In NASIS, "interactive" is used to define the system's ability tc maintain a narrative with the user. This narrative consists of prompts and commands. Prompts are the means by which the system requests data or instructions from the user, while commands are used by the user tc instruct the system on the operations he wishes performed. In addition, NASIS provides a "self-teaching" or "self-instructing" facility to the user. This facility will, upon the user's request, expand upcn a prompt to provide additional definition to the user as to what the system is requesting.

These interactive capabilities of NASIS apply to all of the system's operations and require the use of on-line terminals. Thus, a user, with only basic system skills, could use the majority of NASIS's capabilities with a minimum of training.

B. GENERALIZED SYSTEM

While originally designed to service bibliographical data bases, the current version of NASIS will service a variety of applications. The methods used to define the structure of its data base, to load and maintain information on the data base, and to retrieve information from the data base permit the user to adapt the system to his application without modification. Unique requirements of an application may be satisfied through user prepared validation, conversion, and formatting routines.

C. DATA BASE MANAGEMENT SYSTEM

NASIS, as a Data Base Management System, can best be defined through a description of the following basic functional components of any data base management system:

- A data base structure.
- A means of inputting information to the data base.
- A means of retrieving information from the data base.

The NASIS data base consists of a series of table-driven interrelated files. The files can be classified as linear files, which contain the information; index files, which point to the information; and a descriptor file, which defines all of the files in the data base. Through its Maintenance Subsystem, NASIS provides the user with the facilities to interactively define the structure of the data base and its content.

From the input point of view, NASIS provides the user with the facility to perform "batch" maintenance of the loaded data. This facility is provided through the Maintenance Subsystem.

The Retrieval Subsystem of NASIS permits a user to interactively search his data base for desired information, and, when located, to generate resultant output. In addition, the user may define the format in which the output is to appear.

D. SUPPORTING SYSTEM COMPONENTS

To permit the efficient and orderly operation of NASIS in an operational environment, NASIS includes

additional components. These components and the operational environment are discussed below.

NASIS is designed to operate in a virtual or real memory environment on an IBM 360 or 370 Operating System. Its programs are primarily written in PL/I with its utility modules being written in assembler.

The NASIS MT/T (Multi-Terminal/Tasking) Monitor permits the system to run as a multi-terminal task. In effect, this Monitor turns NASIS into an operating system by serving as the first level interface between OS and NASIS, and the user and NASIS.

Centralized logical access to the data base for the mainline programs is provided in NASIS by its Data Base Executive. In addition, the Executive protects the integrity of the data base for its users.

The terminal plays an important part in the user's interface with the system. This interface is controlled by the Terminal Support Subsystem.

E. SUMMARY

Schematically, the interrelationships of these components are represented by Figure 1. The remaining sections of this document describe these components in more detail. Also discussed are the statistical and utility facilities of the system.

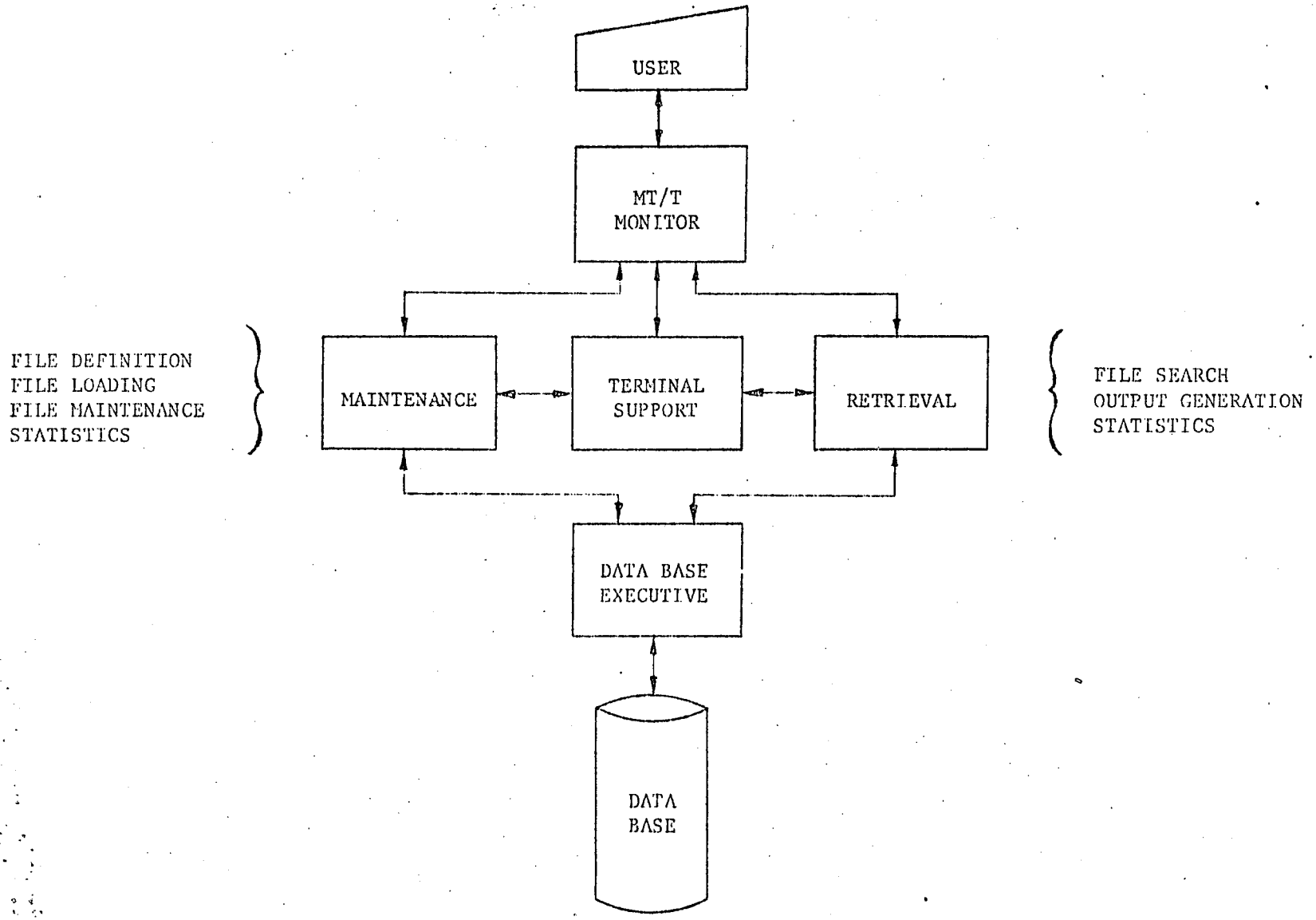


FIGURE 1. SYSTEM OVERVIEW

II. SUPPORTING SYSTEM COMPONENTS

There are components of NASIS which support the operation of the mainline programs. The following components are described in this section:

- MT/T Monitor
- Terminal Support Subsystem
- Data Base Executive

This section also defines the command language with which the user controls and instructs NASIS.

A. MT/T MONITOR

NASIS was programmed under OS which had no provisions for multiple users in a single region. Requirements in the areas of system response and administrative control, on the system when it is used by multiple users, are avoided in NASIS through its MT/T Monitor which permits the system to run as a Multiple Terminal/Task (MT/T) application.

Internally, the MT/T Monitor "sits on top of" the remainder of the system and provides the first level interface between NASIS, the user and OS. It provides the following operating system functions:

- Multi-programming

- Time slicing
- User-to-user communications
- Terminal read/write requests
- Sub-task scheduling

From the external point of view, the MT/T Monitor enhances the administrative control of the system by providing the follcwing:

- Simplified operator-user communications
- Improved system scheduling
- Improved system usage monitoring

B. TERMINAL SUPPORT

Serving as the interface between the MT/T Monitor and the mainline programs is the Terminal Support Subsystem. Generally, this subsystem facilitates the transfer of information between the user and NASIS, and provides an efficient and orderly method of tailoring the appearance of NASIS to its users.

Specifically, the Terminal Support Subsystem provides the following facilities:

- Command and data prompting, allowing the user to direct and control the operations of the system.
- Attention/interrupt processing, allowing the user to initiate, terminate or temporarily suspend NASIS activities.
- Display formatting, including dynamic definition of terminal display dimensions, and "fitting" output to the physical constraints of a particular terminal. Thus, NASIS is independent of physical terminal peculiarities.
- Profile management which permits a user to define his use of NASIS. The contents of the profile are controlled by the user, and are referenced by NASIS during its operation to properly determine an operating environment.

To provide these facilities, an extension to the PL/I language was developed. This extension, entitled TSPL/I, permits the use of PL/I source statements for

performing I/O operations on the terminal. TSPL/I provides the programmer with these facilities:

- GET, PUT, READ and WRITE facilities much like those used in normal data management.
- Terminal facilities for handling attention interrupts, and hardware/software errors.
- PROMPT facility.

The PROMPT facility permits the use of a "message" file from within PL/I programs. This file consists of messages for use by the system in its communications with users. The message file may be tailored by the Data Base Administrator to meet specific needs.

To support TSPL/I, a compilation time source program processor has been developed as a preprocessor of the TSS PL/I compiler. The processor analyzes the TSPL/I statements and generates, in their place, ordinary PL/I statements. If the interface with the mainline programs is respecified, the processor may be changed appropriately to generate suitable statements. The mainline programs may then be recompiled without being reprogrammed.

C. DATA BASE EXECUTIVE

The Data Base Executive Subsystem serves the following functions:

- It protects the integrity of the data base for its owner.
- It provides logical access to the data base as a centralized service for the mainline programs.

These functional aspects of the Data Base Executive are described in greater detail in the following paragraphs.

C.1 Data Base Integrity

The Executive allows the owner complete access to his data base. It protects the data base from other users, but allows the owner to share a data base with other users. However, this data base "sharing" is on a read only basis. This type of security is available on the data base, file (linear only), record and field level.

The Executive further protects the data base by

maintaining the following controls:

- Precludes concurrent maintenance on an individual file.
- Does not allow retrieval on a partially created file.
- Disallows maintenance during reorganization and vice-versa.
- Allows maintenance concurrent with retrieval.

Inadvertent destruction of the index files is precluded by allowing "read-only" access to these files. This is made possible by the Executive's ability to completely maintain the index files.

On the field level, the Data Base Executive aids in data base protection via specific user defined validation and/or conversion routines, and security classifications.

C.2 Data Base Access

The Executive Subsystem provides access to the data base for all mainline programs. These access requirements necessitated the development of an extension to the PL/I language, called DBPL/I. The mainline PL/I programs have ordinary PL/I input/output statements for non-data base I/O. However, for all data base I/O such as variable-length VISAM support, variable-length multi-element field access, operations on lists of keys, etc., DBPL/I statements are used. Such DBPL/I statements cause the transfer of data between the data base and mainline programs. These DBPL/I statements are similar in meaning and form to ordinary PL/I I/O statements so as to be learned and used easily by the mainline PL/I programmers.

Through DBPL/I statements, the mainline programs call the Data Base Executive, as a sub-routine, to perform the following functions:

- OPEN, CLOSE or ERASE files.
- Locate a work area for a new record, or read an existing record.
- Validate, store or replace field elements.
- Extract field elements and transform them for output to the mainline.

- Retain lists of keys in main storage for retrieval purposes.
- Combine lists of keys through its LIST subroutine.
- Determine the number of elements in a list of keys through its #LIST subroutine.
- Free lists of keys when they are no longer needed in main storage.

As with TSPL/I, a preprocessor has been developed to support DBPL/I.

D. COMMAND LANGUAGE

NASIS is an interactive system which maintains a "narrative" with the user. It is through the command language that the user "instructs" the system on the operations he wishes it to perform. The command language is divided into the following categories:

- System Commands
- Subsystem Commands
- Immediate Commands

The commands for each of these categories are described in the following narrative.

D.1 System Commands

The system commands, which control the overall operation of the system are as follows:

- MAINTAIN initiates the maintenance subsystem for the user. This is a stand-alone procedure.
- RETRIEVE initiates the retrieval subsystem for the user.
- RERUN instructs the system to begin retrieving commands from the stored strategy. Strategy is defined as the string of commands issued by the user to accomplish a particular objective.
- RESTART permits the user to restore the system to the point in his strategy that was being executed when his session was abnormally terminated.

- SECURE permits the user to specify alternate security codes.

D.2 Subsystem Commands

The subsystem commands are used with the maintenance, retrieval, statistics, and utilities subsystems. These commands are used to initiate the facilities of these subsystems, and are discussed in the sections of this document describing the subsystems.

D.3 Immediate Ccmmands

The immediate commands are used to provide capabilities useful at all levels and within all subsystems. These commands may be entered whenever the system is waiting for data to be entered. The immediate commands may be grouped into the general functions of message handling, profile services, strategy services, and system control.

The message handling facilities of NASIS provide a "self-documenting" or "self-teaching" capabilitiy. The commands used to provide these capabilities are as follows:

- The EXPLAIN command is used to request from the system an explanation of a term, a message, the origin of a message or the valid responses to a message. This information is obtained from the system's message library which is maintained by the Data Base Administration.

- PROMPT is used to display the message specified by the user on his terminal.

The profile services of NASIS allow the user to streamline the use of the system. The commands providing these services are as follows:

- SYNCNYM adds a new temporary entry into the list of command synonyms. Subsequently, the user will be able to invoke the command by entering its true name, its synonym, or an abbreviation for either.

- DEFAULT adds a new temporary entry into the list of keyword defaults. Subsequently, any defaulted reference to the keyword specified will be replaced by the value specified in the most recent DEFAULT command.

- PROFILE takes the synonym and default values

specified during the current terminal session and make them a part of the user's permanent profile.

- SYNONYMS displays the current synonym values for the terms specified. The user may request one, a list, or all synonym values.

- DEFAULTS displays the current default values of the keywords specified. The user may request one, a list, or all default values.

The strategy services of NASIS are used to save, for future use, the relevant commands entered by the user during his terminal sessions. The STRATEGY command displays either the names of strategies stored in the strategy library, or the contents of a particular strategy, or deletes a strategy from the library.

The remaining immediate commands are used to control the flow of the system through the various system and subsystem levels, to control the input data mode, to page through multiple page displays, etc. The commands used to provide these capabilities are:

- GO continues processing from the point of an attention interrupt.

- END terminates the current command string by raising the END condition. In this way, any program may be terminated at any time, and may provide provisions for a logical exit by performing any "clean-up" operations required.

- APOFF terminates the user's current session.

- KA specifies that the full EBCDIC character set is to be used when reading and writing to the terminal. This permits the user to use upper and lower case data if he so desires.

- KB specifies that the folded EBCDIC character set is to be used which automatically translates lower case alphabetics to upper case. This permits the user, for convenience, to enter his terminal input in lower case.

- PAGE changes the current screen image to another screen image within the same display. The user has the option of paging either forward or backward.

III. DATA BASE STRUCTURE

NASIS revolves around a data base which consists of a series of interrelated, table-driven files. While many data bases may exist under NASIS control, a single data base is the largest operaticnal unit of NASIS.

Normally, a data base consists of the following interrelated files:

- Anchor file
- Associated file(s)
- Subfile(s)
- Index file(s)
- Field descriptor file

These files, in turn, consist of defined units of logical information. The lowest unit of logical information in a data base is an ELEMENT. A FIELD, the next higher level, may contain multiple elements. Fields are grouped to form RECORDS. Each file in a data base is comprised of multiple keyed records.

This section will define the interrelationships of these files and their data content. For additional clarification on the structure of a data base and the interrelationship of the data content, Figures 2 and 3 have been included.

A. LINEAR FILES

The anchor, associated, and subfiles of a data base are linear. Either anchor or associated files may contain any field in the records; however, the information in these files is mutually exclusive except for the key. The records of these files are physically separate, yet they are logically one record.

The subfiles are logical extensions of the anchor or associated files which permit the user to have multiple subrecords per parent record. The parent file record "points" to its related subfile records through a "control field" on the record, which contains the keys of all related subfile records. Similarly each subfile record contains the key of its "parent" record.

B. INDEX FILES

As many as sixteen index files may exist within one data base. The key of an index record is a field, other than the key, on the linear records. The balance of an index record will consist of the keys of all linear records relative to the index key.

The index files are automatically created and maintained by the system in order to assure their completeness and accuracy. This feature is made possible through the field descriptors described below.

C. FIELD DESCRIPTORS

To avoid having to define a particular file in each program using it, each file in the data base is described by a set of field descriptors. A field descriptor defines whether the field should be in anchor, associated, or subfiles, and whether or not the field is to be indexed. Additional attributes are:

- Name
- Type (alphanumeric, etc.)
- Format (fixed length, variable length, etc.)
- Exit routine (validation, conversion, formatting, validation agruments)
- Field security

While the descriptors for the associated, index, and

subfiles reference only the fields contained within them, the descriptors for the anchor file are more comprehensive. In addition to defining the fields contained within the anchor file, the anchor descriptor file also contains dummy field descriptors which reference the fields contained in the other files. In a sense, the anchor descriptor file serves as the data base descriptor file.

These descriptors serve as the source of all information required by the Data Base Executive to perform I/O operations and validation on a data base.

D. UNIVERSAL RECORD FORMAT

To promote standardization and uniformity of the data base components, a UNIVERSAL RECORD FORMAT has been defined. This format is used for every record in a data base, and is illustrated by Figure 4. Each record is divided into fixed and variable portions, each containing the fixed-length and variable-length fields respectively. In the fixed portion, fields are accessed by the high-order position, while the relative position and field lengths are used to find the variable-length fields.

The fixed portion is the first part or base portion, of each record, and it, in turn, is composed of three logical parts. The first part is the TSS maintained field indicating the length of the entire record. The second part is the record's key (unique identifier). The remaining fixed-length fields constitute the third part.

The remainder of each record, following the base portion, consists of the variable-length fields. To locate these fields, a relative field number, relative to the first variable-length field, is defined within the field descriptor. Since each variable-length field is preceded by a length indicator, a variable-length field can be accessed by using the field's relative number in conjunction with the appropriate field lenghts. Null variable-length fields consist of the length indicator itself to preserve field position.

Some fields within the variable portion may be composed of multiple identically formatted elements. These elements may be either fixed-length or variable-length. The recognition of the proper element, for maintenance purposes, is achieved by referring to the proper element by value. All variable-length elements are preceded by a length indicator which is not retained in the case of null elements.

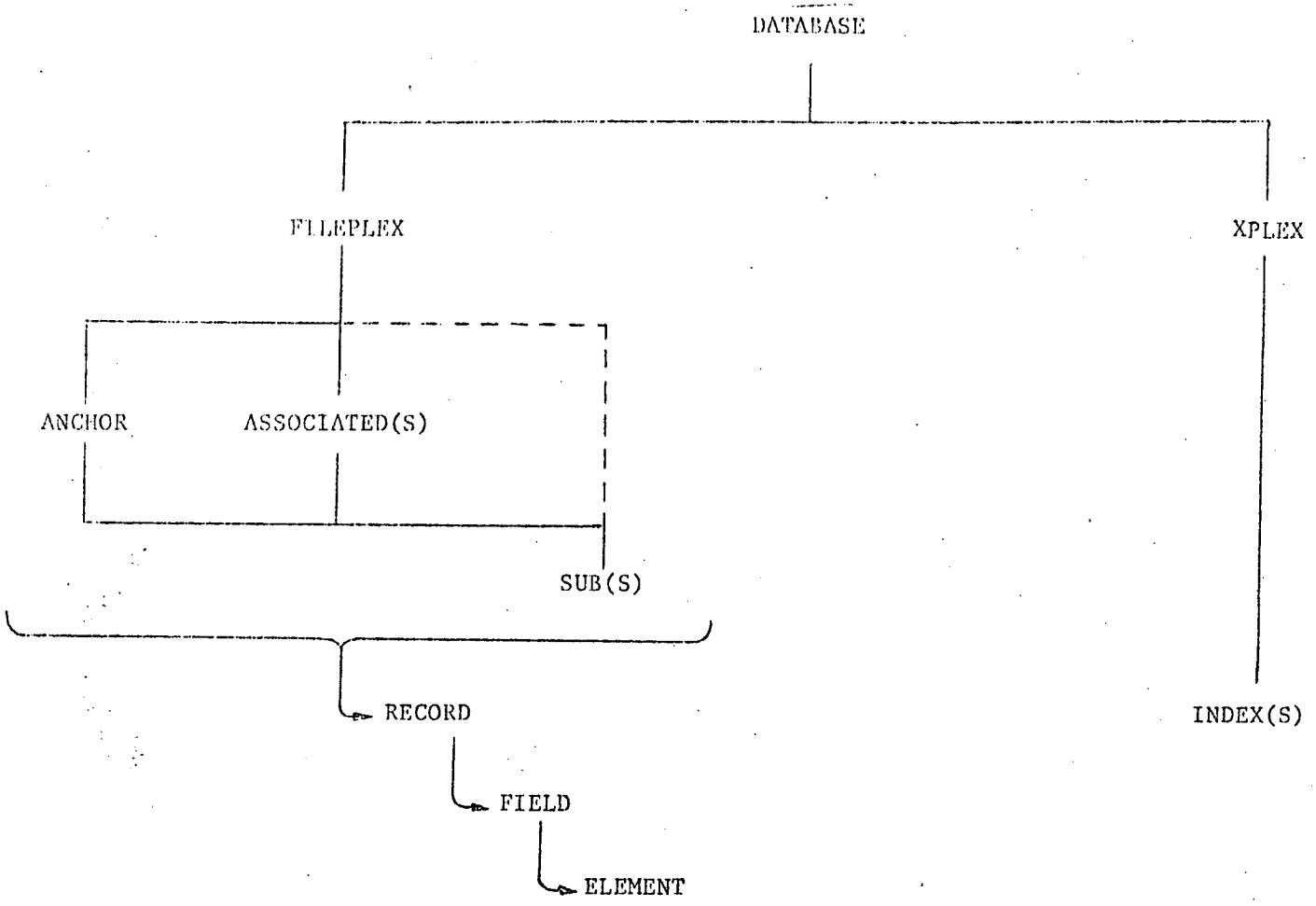


FIGURE 2. DATA BASE STRUCTURE

FILEPLEX

ANCHOR

KEY	FIELD 1*	CONTROL FIELD	FIELD 3
KV1	F1V1	SK1	F3V1
KV2	F1V1	SK3 SK4	F3V2
KV3	F1V3	SK2	F3V2
KV4	F1V2	SK5	F3V4
KV5	F1V1	SK6 SK7	F3V3

ASSOCIATED

KEY	FIELD 4	FIELD 5*
KV1	F4V1	F5V4
KV2	F4V2	F5V2
KV3	F4V1	F5V4
KV4	F4V3	F5V1
KV5	F4V1	F5V4

SUB

S/F KEY	PARENT KEY	FIELD 6*	FIELD 7
SK1	KV1	F6V3	F7V1
SK2	KV3	F6V1	F7V2
SK3	KV2	F6V5	F7V3
SK4	KV2	F6V2	F7V4
SK5	KV4	F6V7	F7V5
SK6	KV6	F6V1	F7V6
SK7	KV7	F6V3	F7V7

* INDICATES INDEXED FIELD

XPLEX

FIELD 1 INDEX

INDEXED FIELD	KEYS	
F1V1	KV1	KV5
F1V2	KV2	KV4
F1V3	KV5	

FIELD 5 INDEX

INDEXED FIELD	KEYS		
F5V1	KV4		
F5V2	KV2		
F5V4	KV1	KV3	KV5

FIELD 6 INDEX

INDEXED FIELD	KEYS	
F6V1	SK2	SK6
F6V2	SK4	
F6V3	SK1	SK7
F6V5	SK3	
F6V7	SK5	

DATA BASE

FIGURE 3. DATA INTERRELATIONSHIPS OF A DATA BASE

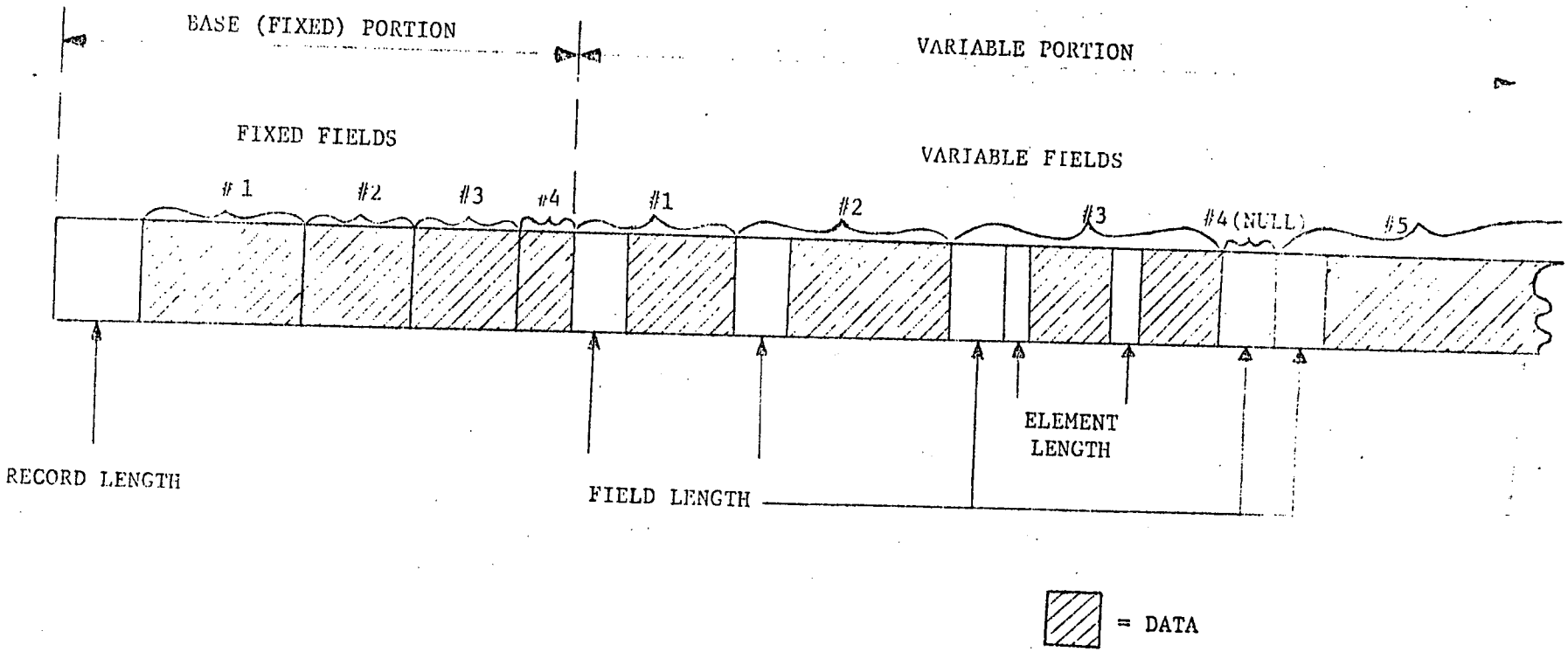


FIGURE 4. UNIVERSAL RECORD FORMAT

R1

IV. DATA BASE MAINTENANCE

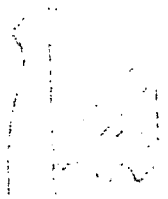
The Maintenance Subsystem of NASIS, represented by the schematic included as Figure 5, provides the user with the following capabilities:

- Data base definition
- Data base loading
- Data base Maintenance

These capabilities are described in this section through their related commands. The usage of these commands is limited to the individual designated as the "owner" of the data base.

A. DATA BASE DEFINITICN

A data base is defined by its field descriptor file. This file is created and maintained by the user, in either a ccnversational or non-conversational mode, using the Descriptor Editor. It is invoked using the EDIT stand-alone procedure. Within the Descriptor Editor there are two modes of operation - CREATE and UPDATE.



A.1 CREATE Mode

This mode is used to create a new descriptor file, or to continue creating an incomplete descriptor file. Using this mode the user can interactively specify the field structure of a dataplex. Within the CREATE mode the user can perform any of the following actions:

- Add a new descriptor.
- Change a previously defined descriptor.
- Create descriptors for an index.
- Create descriptors for a subfile.
- Define field and record security.
- List the names of all existing field descriptors.
- Display a formatted description of a particular field entry.
- Delete a field descriptor.
- File the descriptors.

When defining a field, the user must enter a number of coded or numeric items to describe the type of field being defined. The user may enter as many of these parameters as he desires along with the field name. The Descriptor Editor will prompt the user for missing parameters.

A.2 UPDATE Mode

The UPDATE mode is used to perform maintenance against a specific field descriptor. A field descriptor can be changed in any manner not altering the CREATEd dataplex structure.

B. DATA BASE LOADING

After descriptor creation, the user may load the data base. The normal procedure used to create the NASIS data base is to use the existing linear files of another installation. These "parent" files are processed by the data base loading facilities of NASIS to create the user's data base. These facilities are initiated by the user through the procedures LOAD, INVERT and COMBINE which are described in the following paragraphs.

B.1 LOAD Procedure

To resolve the differences particular to each individual implementation of a file, the facilities provided through the DBLOAD program are generalized in nature. As with any similar generalized approach the primary differences are as follows:

- The nature and format of the input.
- The name of each field of the output.
- The location of the input data element used to create the cutput.

These differences are resolved within the facilities provided through the DBLOAD program, with the exception of an exit routine which must be written by the user. This exit provides the LOAD facility the routines necessary tc do any editing and/or data manipulation desired, and to perform the necessary correspondence between field name and data.

In addition to its primary function of creating the data base, the LOAD facility also provides the user with the ability tc perform the following functions:

- Segment a lcad by restarting at any point.
- Handle errors in a logical and comprehensive manner.
- Accumulate and post statistics.
- Produce an exception-oriented audit trail.

B.2 INVERT Procedure

The creaticn of the index files for a data base may be done as part of the LOAD procedure. However, the most efficient method cf creating or adding index files is provided through the use of the INVERT programs.

This facility permits the user to invert his linear files after they have been loaded, utilizing an OS sort utility and Assembler I/O routines to create the index files. This facility may also be used to invert a field on the linear file once the data base has been established.

B.3 COMBINE Procedure

The COMBINE procedure provides the user with a facility

to combine like index files. This facility is used when indexes beccme too large under normal updating procedures. In such a case, the user would use the INVERT facility to create a temporary index file which would then be merged with the existing index file using the COMBINE facility.

C. DATA BASE MAINTENANCE

Maintenance is the addition of new records to the data base, the deletion of existing records from the data base, and the modification of existing records in the data base.

The user may accomplish the maintenance of his data base using any one of the following methods:

- Following the same method used to create his data base, but using the LOAD facility in the update mode.
- Using the CORRECT facility provided in the Retrieval Subsystem.

The first method is self-explanatory and is used to handle major additions to the files, such as monthly updates. The remaining method involves using the CORRECT command and UPDATE procedure which are described in the following subsections.

C.1 CORRECT Command

The CORRECT command provides a user with the ability to propose changes to any data base permitted to him during a normal terminal session. This command is available to the user in the Retrieval Subsystem.

In the Retrieval Subsystem, the CORRECT command creates a transaction which is stored on the transaction data base. The DBA reviews the transaction data base and decides if the transactions should be included in the maintenance processing. This step is important and necessary in order to prevent the inadverdent destruction of data. These maintenance transactions are then processed by the DBA, in a batch mode, against the master data base.

The transaction data base is created using a TSS utility entitled USERJOIN. This utility is used when the user first "joins" the system. Within the CORRECT command, there are several sub-commands. These sub-commands are used for program control or to indicate the desired maintenance action. Using these

subcommands, the user may propose additions, deletions and modifications to data on an existing data base. If the user indicates such, the system will generate a verificatiсn display showing what the record would look like once changed.

C.2 UPDATE Procedure

The Maintenance mainline program processes the transactions created with the CORRECT command against the data base. As each transaction is processed, it is deleted from the TRNSCT data base.

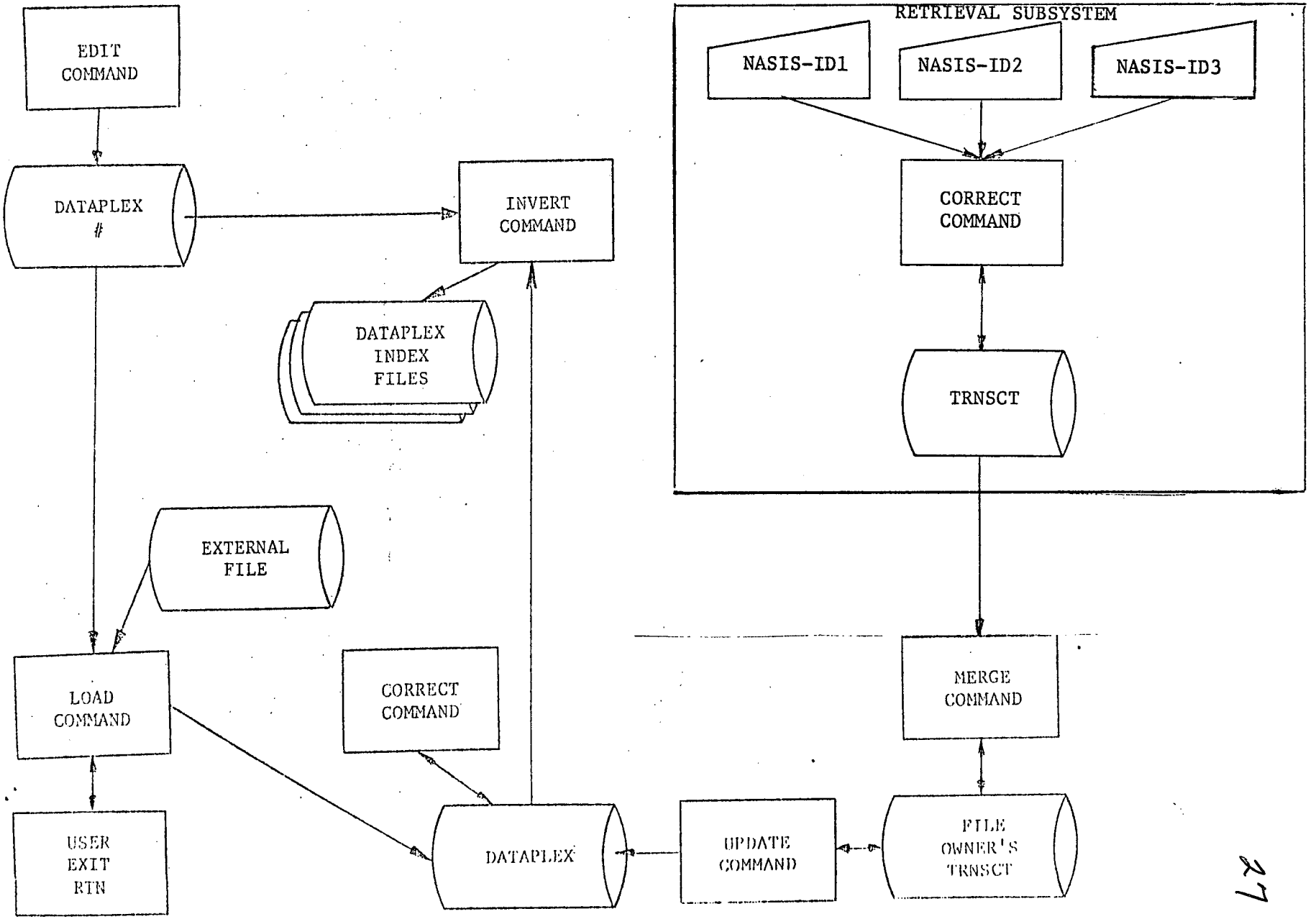


FIGURE 5. MAINTENANCE SUBSYSTEM OVERVIEW

V. DATA BASE RETRIEVAL

The NASIS Retrieval Subsystem is invoked by the RETRIEVE command. The facilities provided by this subsystem permit the user to (1) search a data base, and (2) generate output.

This subsystem can be operated in either a conversational or non-conversational mode. In the conversational mode, the user interacts with NASIS via a terminal. In the non-conversational mode, the subsystem serves as a form of job control language, since no interactive monitoring is performed by the user.

The following subsections describe these facilities along with the related command language.

A. SEARCH

The SEARCH facilities of the Retrieval Subsystem provide the user with the capability to "browse" through a data base using its index files, to select and combine sets of pertinent record keys, and to preform a linear search on those keyed records. The sequence of commands used to initiate these actions is referenced as a search strategy. A strategy once defined, may be stored for future usage.

As with the other NASIS facilities, these facilities are initiated by the user via the command language. The commands related to the search function are FIELDS,

EXPAND, SELECT, SEARCH, SETS, EXECUTE and CANCEL. Following is a description of these commands.

A.1 FIELDS Command

To obtain the names of all fields in a data base, the user enters this command. In addition to listing the field names, the system indicates those that are indexed.

A.2 EXPAND Command

This command permits the user to "browse" through the index files of a data base. The user accomplishes this by entering, along with the command, the name of the indexed field and a data value (a field content). The system responds with an EXPANDed display of the index. The display shows the designated data value and the values beyond it. In addition, the number of anchor keys linked to each data value and a reference (E) number are displayed.

A.3 SELECT Command

The SELECT command takes as input an expression describing the user's criteria for selection of records from the data base. The final result is a list of record keys (called a "set") pointing to those records which meet the stated criteria.

In the simplest case, a SELECT expression may be only a value specified for a particular field. However, Boolean combinations of values, previously formed sets (set #), EXPAND line numbers (E#), yet to be done searches (S#), and ranges of these terms may also form a SELECT expression.

References to an indexed data base fieldname causes the SELECT command to generate a set immediately. Fields which are not indexed require a linear, record-by-record, search to satisfy the user's criteria and generate a set of keys. Searches are coded and saved until the EXECUTE command is entered. At this time all search entries are evaluated.

A.4 SEARCH Command

The SEARCH command is used to enter linear search expressions which are then passed to the SELECT command. It's purpose is to make it easier for the user to use the linear search facility of the command system.

A.5 SETS Command

The user may review the sets or pseudo-sets he has created so far in a strategy session by entering the SETS command. The user has the options to (1) review all his sets, (2) review all sets from a particular set number forward, (3) review all his pseudo-sets, or (4) review all pseudo-sets from a particular pseudo-set forward.

A.6 EXECUTE Command

The EXECUTE command is used by the user to execute his linear search strategy and convert his pseudo-sets to actual sets of anchor keys. The linear search may be executed in either conversational or background mode.

A.7 CANCEL Command

During a retrieval session, the user may specify actions to be performed later, either later in the session, as in the case of a search, or later, after the session is complete, as in the case of prints. If, for some reason, the user decides that these actions are not necessary, he may use the CANCEL command to eliminate them. The CANCEL command can be used to delete all pseudo-sets and PRINTs on pseudo-sets or to delete normally queued PRINTs.

B. OUTPUT GENERATION

The output generation facilities of the retrieval subsystem provide the user with the capability to either display his retrieved data at his terminal (or SYSOUT listing if the task is non-conversational), or to have it printed on the high speed printer. The user may use one of the predefined formats for his output, or he may define his own format.

The predefined formats avaiable in NASIS are identified by a one digit number and are as follows:

- "1" The anchor or subfile keys of a particular set.
- "2" All fields on the anchor file for a particular record.
- "3" All fields on the anchor and associated files for a particular record.
- "4" All fields on the anchor, associated and subfiles for a particular record.

- "5" All fields on the subfiles for a particualr reccrd.

The user defined formats may be either sequential or columnar and may be stored for repeated usage.

The commands providing these facilities to the user are FORMATS, FORMAT, DISPIAY, and PRINT. These commands and their functions are discussed in the following subsecticns.

B.1 FORMATS Command

The FORMATS command displays for the user a list of all output fcrmats available. The list will be alphabetically seqvenced, and predefined formats will be distinquished from the stored user defined formats. The user defined stored formats will be identified by the name, a number preceded by an "F" if they are columnar, or a number greater than "5" if they are sequential.

B.2 FORMAT Command

This command is used to define a customized format for later usage with the DISPLAY or PRINT commands. With this command the user may define a new sequential or columnar fcrmat, or revise an existing format.

Within the FORMAT command the following sub-commands are available:

- DISPLAY which simulates the effect of using the DISPIAY retrieval command with the current format.
- END which terminates processing of the current format and returns the user to the retrieval commands.
- FIELD which permits the user to add, delete or revise the field specifications of the current format.
- HEADER which allows the user to specify new or revised cclumn headings on one line at the top of each page of output.
- NAME allows the user to assign a "name" to a format previously identified by a number.
- STORE is used to "store" a format for future use. In addition, it may be used to assign a

"name" to the format.

- TITIE is used to specify a title for the top of each page of output.

B.3 DISPLAY Command

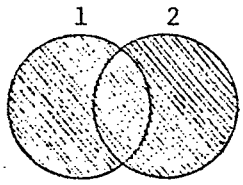
The DISPLAY command is used to output to his terminal or SYSOUT listing (if non-conversaional) one of the following data sets:

- A data record according to a specified format.
- A set of keys, thus data records, according to a specified format.
- A specific field of a data record.

B.4 PRINT Command

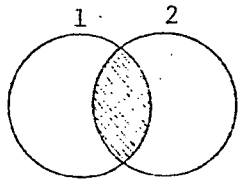
This command is functionally the same as the DISPLAY command. The exception is that the output will be printed on the high speed printer.

"OR" OPERATOR



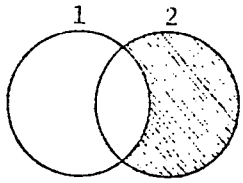
1 + 2

"AND" OPERATOR



1 * 2

"NOT" OPERATOR



2 - 1

INDICATES RESULTING SET.

FIGURE 6. BOOLEAN OPERATORS

VI. USAGE STATISTICS

To provide the user with the information he requires to effectively monitor and control the usage and status of the system and data bases, a usage statistics capability has been included in NASIS. The statistics provided by this capability, which is essentially a separate subsystem, are divided into two groups - retrieval or usage, and maintenance or status statistics. These statistics are automatically maintained by NASIS and are accumulated on a statistics file.

These two statistical breakdowns, and their related commands and outputs are discussed in the following narrative.

A. RETRIEVAL STATISTICS

The retrieval statistics include counts on the number of times that various commands have been invoked, the number of retrieval sessions, the dates and time used for those sessions, and the aggregate time spent retrieving data.

Two outputs are available to the user on retrieval statistics. The session statistics are generated when the counters are re-initialized, or checkpointed. The checkpoint report, for a NASISID, contains a line entry for each of the sessions on record. These line entries display the command counts, the lines, the date, the data base name, and other pertinent information. This report will permit the Data Base Administrator to

analyze the usage that a user is making of the system and of the individual data bases.

The second report is generated by the command REPORT. The content of this report is similar to the checkpoint report. Also included is a summary of the aggregate times and sessions for all users.

B. MAINTENANCE STATISTICS

The maintenance statistics include counts of the number of records added, deleted or replaced on the anchor, associated, sub and index files. A count of the total records on the anchor file is included. These statistics are available only to the owner of a data base through the maintenance statistics report.

This report may be initiated through the maintenance subsystem using the PRINT command.

These maintenance statistics will relay to the Data Base Administrator the status of his data base, and will assist him in determining when to backup the system, when to reorganize his files, and other items related to the control of the system.

VII. UTILITIES

Through the Utilities the following capabilities are made available tc the NASIS user:

- USERID file maintenance

These capabilities and the commands which initiate them are discussed in the balance of this section.

A. USERID FILE MAINTENANCE

The ability to create and maintain the NASIS.USERIDS file is prcvided to a Data Base Administrator through the JOIN command. This data set contains the identification ccdes under which the users of NASIS are given access to MT/T, the retrieval subsystem, and the various data bases. The key of each variable-length record in the data set is the identification code. The remainder cf each record contains the user's password, time slice, user authority, and a list of the data bases to which he has access.

APPENDIX A.

NASIS DOCUMENTATION

For greater detail about the design, structure, and usage of NASIS, the following documents are available:

1. INSTALLATION STANDARDS
2. NASIS OVERVIEW
3. DATA SET SPECIFICATIONS
4. PROGRAM DESIGN SPECIFICATIONS
5. RETRIEVAL COMMAND SYSTEM REFERENCE MANUAL
6. SYSTEM MESSAGES
7. DBA REFERENCE MANUAL